scientific reports

Check for updates

OPEN

Screen shooting resistant watermarking based on cross attention

Lianshan Liu[1]✉, Peng Xu[1] & Qianwen Xue[2]

With the development of digital imaging devices, the process of recording sensitive information displayed on screens through mobile phones and cameras has become a prominent technique for modern data leaks. In order to identify the origin of information violations, Screen-Shooting Resistant Watermarking (SSRW) has attracted a lot of attention. Most existing solutions are based on Convolutional Neural Networks (CNNs) for the embedding of watermarks. However, due to the limited reception field of CNNs, they are proficient in extracting local features but cannot understand the entire image. This paper presents a new watermarking system that is resistant to screen recording, with multi-head and cross-attention to incorporate watermarks, replacing the encoder in the end-to-end architecture. Specifically, we segment the image and watermark into smaller patches for positional embedding. Afterward, we calculate the attention scores through multi-head attention layers and generate the encoded image through concatenation. This approach increases the model's ability to comprehend the entire image, thereby increasing performance. In addition, we enhance the U-Net network structure to replace the end-to-end decoder. The experimental results demonstrate that the proposed method not only reaches more than 95% accuracy in different capture scenarios but also excels in terms of reliability and invisibility relative to current state-of-the-art (SOTA) methods. In addition, this approach yields impressive PSNR and SSIM average values of 41.90 dB and 0.99, showing the excellent visual quality and reliability of the watermarked images.

Keywords Robust watermarking, Screen-shooting, Deep learning, Cross attention

In the digital age, with the rapid development of information technology, protecting the copyright of digital content has become increasingly important. Image steganography, a technique for hiding information within images, has been extensively studied for its ability to maintain the confidentiality of information[1–8]. Digital watermarking, as a branch of image steganography, focuses on embedding specific information, such as copyright ownership, into digital products like images, audio, and video in a way that is invisible to the human eye or ear. Video watermarking extends this concept by integrating watermarks into video content to ensure copyright protection and content authentication. Similar to image watermarking, video watermarking techniques can be classified into spatial and frequency domain methods, with additional considerations for temporal coherence across frames, as discussed in[9–11]. This embedded information can survive various attacks, such as compression, cropping, and filtering, and can be extracted when necessary to prove the authenticity and ownership of digital content. The techniques used in digital watermarking include spatial domain methods like LSB (Least Significant Bit) replacement and frequency domain methods like DCT (Discrete Cosine Transform) and DWT (Discrete Wavelet Transform). In recent years, the integration of deep learning technologies has emerged as a growing trend, offering new possibilities to enhance the robustness and invisibility of watermarking algorithms.

With the development of the Internet and multimedia technologies, digital imaging devices such as smartphones and cameras have become a common tool for people to quickly acquire information. Individuals can capture information from screens with their phones without leaving any trace. At the same time, screen capture has become an important means of pirating digital media content and stealing confidential information. In this context, digital watermarking, which embeds specific identification information in images in an intangible way, can be used to track the origin of data[12].

Traditional digital watermarking schemes often rely on hand-made features to embed watermarks in the redundancy of the spatial or frequency domain[13]. However, these schemes lack sufficient robustness to deal with screen-shooting processes. This is due to the fact that screen-shooting is a process of transmedia transmission

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China. [2]Qingdao Maternal & Child Health and Family Planning Service Center, Qingdao 266520, China. ✉email: liulianshan@sdust.edu.cn

that produces a significant amount of complex distortions that can seriously damage embedded watermarks. Traditional watermarking approaches usually focus on distortions caused by digital editing and electronic channel transmission, without prior knowledge of transmedia distortions. Therefore, it is urgently necessary to develop a watermarking algorithm that can withstand the distortions introduced by screen-shooting.

Recently, as deep learning progresses, many researchers have used the deep neural network architecture for the insertion and retrieval of watermarks. Deep Neural Networks (DNNs) are able to independently learn and adapt to different watermark requirements and distortions. By incorporating noise layers for end-to-end training, they improve the resistance to certain distortions and provide a stronger and more effective approach to the protection of digital assets. Current deep learning-based watermarking models generally adopt an end-to-end encoder-noise layer-decoder (END) architecture that uses CNNs for feature extraction. The task of the encoder is to embed the watermark into the image, and the decoder is designed to extract the watermark information from the resulting encoded image. By adding noise that simulates screen capture distortions in the noise layer, a certain degree of robustness can be achieved against screen-shooting attacks. In the END structure, the encoder and the decoder are interdependent, with the decoder being essentially the reverse process of the encoder. Therefore, the design of an effective encoder is crucial for the overall performance of the watermark[14].

Recent studies have shown that vision transformers (ViTs)[15] outperform CNNs in tasks that require a holistic understanding of image content. This capability is derived from their ability to gather extensive contextual information through attention mechanisms, as opposed to CNNs, which are expert in extracting local features. Using ViTs' global receptive field, the overall features of the cover image and watermark can be more effectively used, thereby increasing the robustness and invisibility of the watermark. The researchers conducted preliminary investigations into the insertion of watermarks with ViTs ideas. Dasgupta et al.[16] used self-attention and cross-attention mechanisms to introduce watermarks. However, this approach has high computational complexity and long training periods. Furthermore, it is highly sensitive to JPEG compression and is unable to resist screen shooting attacks. Therefore, in this paper, we improved this system by proposing a lighter and more robust watermark embedding methodology. Specifically, we divide images into 8×8 pixels using a sliding window technique and use cross attention with multiple heads for the watermark integration. In addition, we design a dynamic noise layer that simulates screen-shooting distortions and improves the robustness of the watermark. This not only accelerates the training process, but also ensures sufficient watermark strength. Furthermore, we strengthen the U-Net network structure to serve as a decoder.

The contributions of this work can be summarized as follows:

- Proposed a novel lightweight deep learning-based watermarking scheme.
- Designed a dynamic noise layer to simulate screen-shooting attacks.
- Enhanced the U-net network architecture for decoding.

The remainder of the paper is organized as follows: Section "Related work" discusses relevant literature, Section "Proposed scheme" details the proposed watermarking approach, Section "Experiments" showcases the experimental findings and their analysis, and Section "Conclusion" concludes the discussion.

Related work

In this section, we first introduce some traditional methods of implementing image watermarks, then provide a general overview of deep-learning image watermarking schemes, then present some watermarking approaches that show strong robustness against screen capture attacks, and finally discuss the contents associated with Vision Transformers.

Traditional image watermarking

Conventional image watermarking techniques are usually divided into two categories: spatial domain methods and frequency domain methods. Spatial domain methods directly modify pixel values to embed watermarks, while frequency domain methods first convert images to the frequency domain and then modify transformation coefficients to incorporate watermarks. The spatial domain techniques consist mainly of LSB[17] and the patchwork method[18]. Frequency domain methods include techniques based on Discrete Fourier Transform (DFT)[19], DCT[20], DWT[21], and Singular Value Decomposition (SVD)[22]. These methods focus on improving the invisibility or robustness of watermark images after digital editing (e.g. compression, filtering) or geometric transformations (e.g. translation, rotation, scaling). However, they are relatively vulnerable to screen capture attacks. To counter screen capture attacks, Fang et al.[23] introduced an intensity-based scale-invariant feature transform(I-SIFT) algorithm to identify suitable regions for embedding watermark templates. Chen et al.[24] introduced a local region feature extraction method that uses Harris-Laplace and SURF corner point detection for feature synchronization. Amiri et al.[25] Images of the DWT domain were analyzed and a genetic algorithm was used to find suitable locations for a watermark embedding. Huang et al.[26] observed that the spread spectrum and quantization are susceptible to interference of host signals and proposed an adaptive spread spectrum scheme with self-adjustable embedding strength. In summary, traditional image watermarking systems incorporate watermarks by modifying the pixel value in the spatial domain or converting the coefficient in the frequency domain. However, they rely on hand-made features for embedding and extraction, which are specific to the task but lack generality.

Deep learning-based image watermarking

In recent years, a number of deep learning-based watermarking techniques have emerged, following the introduction and development of DNNs. Zhu et al.[27] introduced the "HiDDeN" architecture, the first deep-learning-based watermarking system. This approach simulates different types of noise by integrating a noise

layer between the encoder and the decoder and improves its resistance to different noisy conditions. Chen et al.[28] proposed JSNet, a more realistic network to simulate JPEG compression. They provided a differentiable JPEG compression module for watermarking. Reference[29] has designed a hybrid attack. Specifically, they used a multi-attack layer to apply different attacks in each iteration, and one of the attacks was randomly selected with a certain probability. Liu et al.[30] presented a two-phase deep learning framework. During the initial phase, the encoder and the decoder are trained together in a noisy environment. In the subsequent phase, the encoder pre-trained in the first phase generates images that are exposed to noise attacks. Disturbed images are used to refine the decoder, allowing it to adapt effectively to the dispersion characteristics and improve reliability. Reference[31] created a set of two independent codebook images to improve security through architecture design and key protection. Fang et al.[32] proposed a DNN-based watermarking algorithm that places the template on the R channel and the image's watermark data on the B channel. Palani et al.[33] propose a semi-blind watermarking method using a convolutional attention-based turtle shell matrix for tamper detection and recovery of medical images, effectively ensuring the integrity and authenticity of medical images, but it faces challenges in computational complexity when dealing with complex images. Fang et al.[34] introduced a decoder-based watermarking scheme with a specific architecture of decoder-encoder-noise layer-decoder. They argued that the encoder could introduce unnecessary redundant features in the watermark insertion, which could harm the decoding. These methods address mainly robustness issues related to digital editing distortions (such as cropping and rotation). Due to the lack of appropriate noise layer designs, however, it is difficult to cope with complex distortions encountered in media processing.

Screen-shooting resistant watermarking

The rapid development and widespread use of digital imaging devices have led to issues such as confidential information leaks and intellectual property infringements. To address these issues, screen-shooting resistant watermarking has been proposed, which not only needs to withstand digital editing attacks but also requires stronger cross-media robustness. One approach to achieve this is to incorporate more realistic simulated attacks into the noise layer. The most straightforward method is to use real data instead of a simulated noise layer for training. Wengrowski et al.[35] used 1.9TB of real data to train their network, achieving moderate resistance against screen capture attacks. However, this method requires extensive image preprocessing, making it impractical for large-scale implementation. In 2020, Tancik et al.[36] proposed an end-to-end trainable neural network (StegaStamp) that simulates screen-shooting attacks through "perspective transformation, motion blur/defocus, color adjustment, noise, and JPEG compression." In 2022, Jia[37] and J. Lu[38] each made improvements to StegaStamp. Jia utilized local regions for both embedding and extracting watermarks instead of the entire image, significantly reducing the preprocessing time and enhancing the invisibility and robustness of the watermark. However, the performance of this method is highly dependent on accurate local region identification. In contrast, Lu substituted the CNNs with DWT and inverse DWT for upsampling and downsampling, resulting in more stable features. However, the distortions caused by the screen-shooting process are too complex to be fully simulated. Fang et al.[39] proposed that simulating only the most impactful parts of the screen capture process can achieve good robustness. They designed a simulated noise layer called PIMoG, which mimics lighting, moiré patterns, perspective, and other attacks. To simulate the actual fusion process and produce robust watermarks, Qin et al.[40] developed a deep noise simulation network. In summary, most screen-shooting watermarking methods enhance robustness by optimizing the noise layer, including the direct introduction of realistic distortion data and distortion simulation.

Vision transformers

The initial application of transformers in natural language processing (NLP)[41] led to significant achievements. Due to their superior performance, they are considered to be replacing the dominance of CNNs. However, the computational complexity of self-attention in transformers increases sharply with the rise in the number of image tokens, limiting their effectiveness in high-resolution images. Later, Vision Transformers (ViTs) provided a new approach to image processing. While CNNs rely on local receptive fields, ViTs process images as a sequence of patches, which allows them to capture global dependencies in the data. ViTs have been applied to diverse computer vision tasks, including image recognition[15,42], segmentation[43,44], object detection[45,46], and more. In the last few years, the application of Transformers to digital watermarking has seen a significant increase among researchers. For example, Luo et al.[14] proposed using a soft fusion module instead of direct fusion, by mining long-range correlations between the cover image and the watermark and integrating them using Transformers, to achieve effective watermark embedding. Dasgupta et al.[16] utilized self-attention and cross-attention to embed watermarks, improving watermark robustness by learning invariant domain representations. Palani et al. introduced two methods: the first[47] used a Swin Transformer to generate robust watermark features embedded via Quaternion Dual-Tree Complex Wavelet Transform (QDTCWT), while the second[33] employed a convolutional attention-based turtle shell matrix for semi-blind watermarking to detect and recover tampered medical images. However, these methods primarily address digital editing distortions and exhibit poor robustness against screen-shooting attacks. The limitations of current approaches are summarized in Table 1.

Proposed scheme

Framework overview

The proposed model architecture consists of four main components: an encoder, a discriminator, a noise layer, and a decoder. The encoder embeds the watermark into the cover image, generating the encoded image. The discriminator evaluates if the image has been encoded. The noise layer modifies the encoded image, and in the end, the decoder recovers the watermark from the altered encoded image. Specifically, the cover image I_c and the original watermark W are input to the encoder E. The encoder converts W into a binary image, divides

1. Computational complexity may limit real-time applications[9,10]
2. Challenges in handling Screen-Shooting complex attacks[48]
3. Robustness and invisibility are difficult to balance[36]
4. The capacity of the watermark is small[33]

Table 1. Limitations of the existing system.

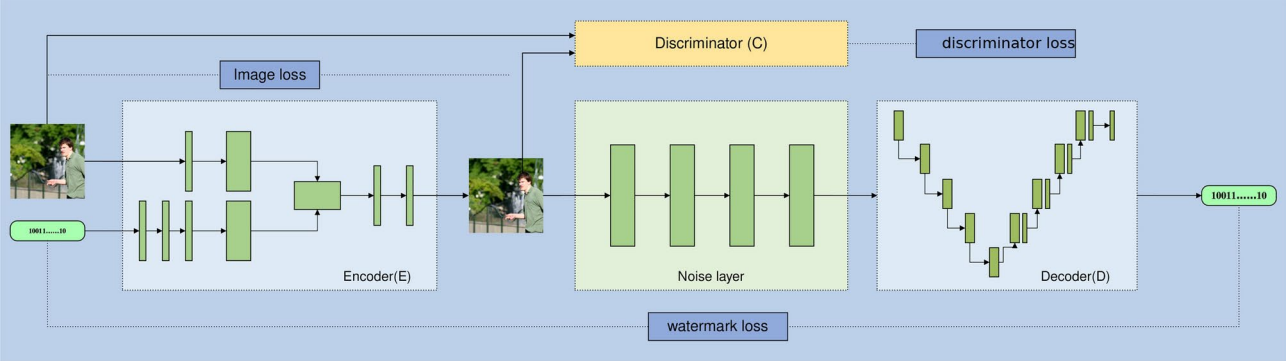

Fig. 1. Framework of the proposed scheme.

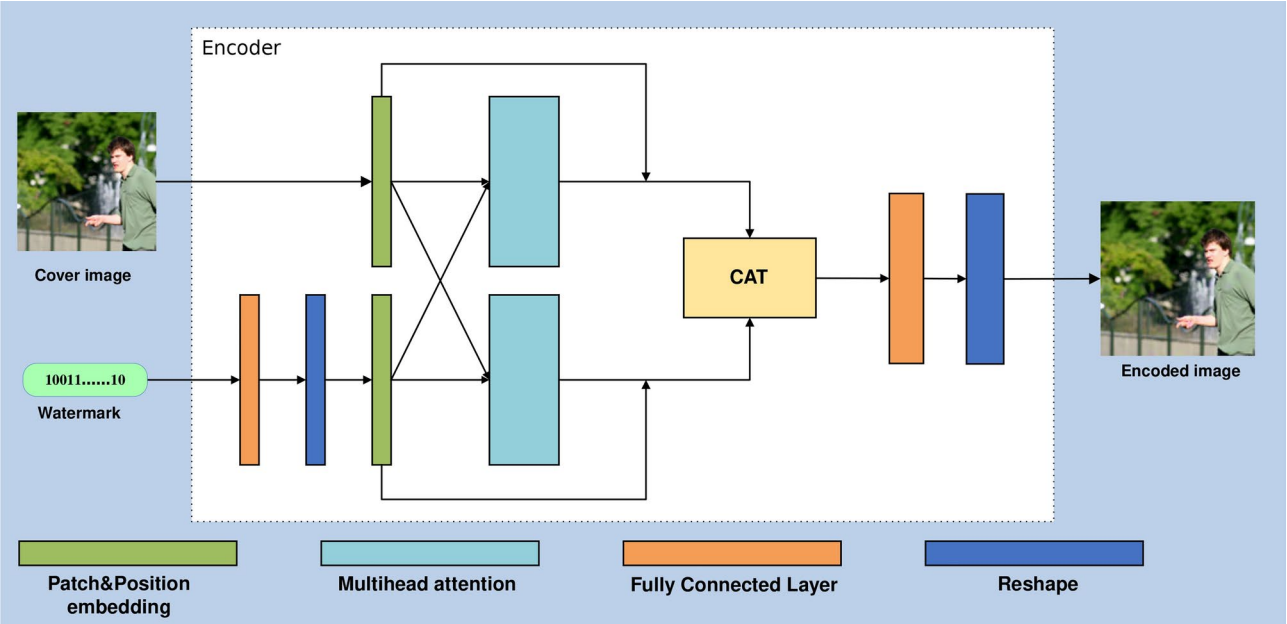

Fig. 2. Structure of encoder.

both I_c and the binary watermark into smaller blocks, and embeds the watermark via a cross-attention layer, producing the watermarked image I_e. The noise layer then applies simulated screen-shooting attacks to I_e, yielding the distorted image I_d. The decoder D, using an enhanced U-Net architecture, extracts the watermark from I_d and outputs the decoded information. The complete architecture is illustrated in Fig. 1. The encoder integrates the watermark into the cover image, generating I_e. After I_e undergoes simulated attacks in the noise layer, the decoder outputs the extracted watermark. The discriminator C evaluates the watermarked image to ensure the encoder produces high-quality outputs.

Encoder

The encoder E is tasked with embedding the random watermark sequence into the cover image. It receives a cover image C of shape $3 \times 128 \times 128$ and a watermark sequence of length 64 bits W, outputting an encoded image M of shape $3 \times 128 \times 128$. The structure of the encoder is illustrated in Fig. 2. We employ cross-attention

and positional embeddings to facilitate watermark embedding, treating the watermark as auxiliary information for its distribution and embedding. Specifically, the 64-bit watermark sequence is first expanded to 256 through a fully connected layer and then reshaped to $1 \times 16 \times 16$. The cover image is divided into 256 patches, each of size 8×8 and the watermark image is similarly divided into 256 patches with 1×1 size. Positional embeddings are incorporated into these patches, which are subsequently analyzed through a multi-attention layer featuring 16 heads. This layer computes attention scores between the cover image patches (serving as queries) and the watermark patches (serving as keys and values), improving the interaction between the cover image and the watermark. These scores help identify the best locations for embedding the watermark. The output of the attention layer is then added back to the original block embeddings, concatenated along the channel dimension, and finally passed through a fully connected layer, with the resulting shape being reshaped to $3 \times 128 \times 128$ as the encoder's output.

The objective of the encoder E is to render the encoded image as similar as possible to the original image. To maintain the visual quality of the encoded image, the blending loss is employed to constrain the encoder. This is expressed mathematically in Eq. (1).

$$\begin{cases} L_E = \lambda_1 L_{l2} + \lambda_2 L_{lpips} + \lambda_3 L_{ssim} \\ L_{l2} = MSE(I_c, I_e) = MSE(I_c, E(\theta_E, I_c, W)) \\ L_{lpips} = LPIPS(I_c, I_e) = LPIPS(I_c, E(\theta_E, I_c, W)) \\ L_{ssim} = 1 - SSIM(I_c, I_e) = 1 - SSIM(I_c, E(\theta_E, I_c, W)) \end{cases} \tag{1}$$

where I_c and I_e represent the cover image and encoded image respectively, and H and W denote length and width of the image.

Noise layer

In an end-to-end architecture, the inclusion of a noise layer for joint training is critical to ensuring watermarks withstand screen-shooting distortions. We inserted a noise layer between the encoder and decoder to simulate distortions typical of the screen-shooting process. Rather than modeling the entire screen-shooting process as a noise layer, we focus on simulating the most impactful distortions to ensure robustness[39]. The specific distortions simulated in the noise layer include perspective, blur, lighting, and moiré distortion, and the resulting attacked image is illustrated in Fig. 3.

Perspective distortion

During screen-shooting process, perspective distortion is almost inevitable because it is difficult for the photographer to ensure that the camera is strictly parallel to the screen. To simulate perspective distortion, we randomly perturb the four corners of the image.

Lighting distortion

During screen-shooting process, lighting distortion is also almost inevitable due to the influence of ambient light and screen backlight. We simulate the screen backlight using a linear gradient light source and the ambient light and reflections using a point light source.

For the linear gradient light source, the gradient mask M can be defined as:

$$M(x, y) = a + b\left(\frac{x}{W} + \frac{y}{H}\right) \tag{2}$$

where a and b are parameters controlling the intensity, and H and W are the height and width of the image, respectively.

For the point light source, we first randomly initialize a point (x_0, y_0) within the image, and the light intensity P at each point is simulated using the following formula:

$$P(x, y) = C(x, y) + i(x, y) \tag{3}$$

where $C(x, y)$ represents the grayscale values of the original pixel channels, and $i(x, y)$ represents the light intensity, obtained from Eq. (4).

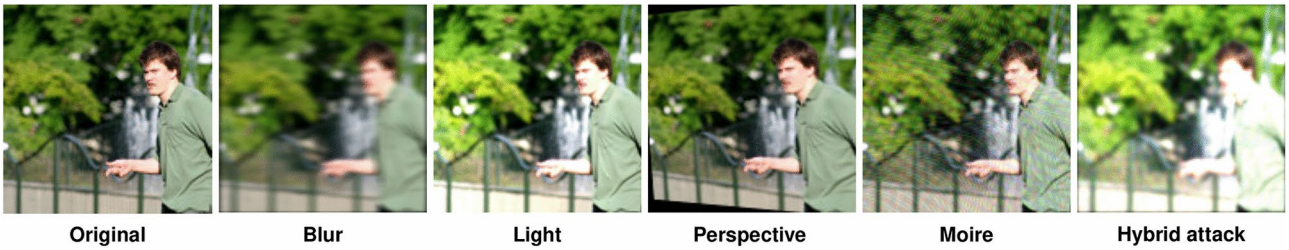

Fig. 3. Images under different attacks.

$$i(x, y) = \frac{i_0}{d(x, y)^2 + \varepsilon} \tag{4}$$

where i_0 represents the initial intensity of the point light source, $d(x, y)$ represents the distance from the pixel to the point (x_0, y_0), and ε is a small constant used to prevent division by zero or producing excessively large values.

Blur distortion

During screen-shooting, two types of blur distortions can occur: defocus blur and motion blur. Defocus blur results from improper focusing, while motion blur is caused by the movement of the lens or the content on the screen. We simulate these two types of blur using a Gaussian blur kernel and a linear convolution kernel, respectively. The standard deviation of the Gaussian blur kernel is randomly sampled between 1 and 3, and the angle of the linear convolution kernel is random, with a width ranging from 3 to 7 pixels.

Moiré distortion

Moiré patterns are a special type of distortion that occur during screen-shooting, caused by the different sampling rates of the lens and the screen. We simulate Moiré patterns using the method described in[39].

Traditional noise layers apply all types of distortions to the image every time. While this ensures robustness against various distortions, it is not realistic because not all types of distortions are introduced in every screen-shooting instance. Therefore, we believe that applying random types and intensities of distortions each time provides better resistance to screen-shooting. We set different application probabilities for the four types of distortions described above, with default settings of 0.9, 0.6, 0.5, and 0.8, respectively.

Decoder

U-Net[49] was initially designed for medical image segmentation tasks utilizing multi-scale feature extraction and fusion, demonstrating excellent performance in the field of image processing. In this work, we modified the U-Net architecture to serve as the decoder D. Specifically, we replaced the traditional down-sampling operation with strided convolutions on the left side of the network and incorporated the original image into feature fusion on the decoder side to compensate for information lost during downsampling. By using strided convolutions for feature extraction and downsampling, we enhanced the decoder's feature extraction capabilities while improving training efficiency and model performance. To compensate for the loss of original features due to large-stride convolutions, we downsampled the original input image at various scales and merged these downsampled images with the upsampled sections on the right side, thereby further enhancing the feature extraction capabilities of the decoder. The modified U-net architecture is illustrated in Fig. 4.

The objective of the decoder D is to minimize the difference between the decoded watermark sequence and the original watermark sequence by updating θ_D:

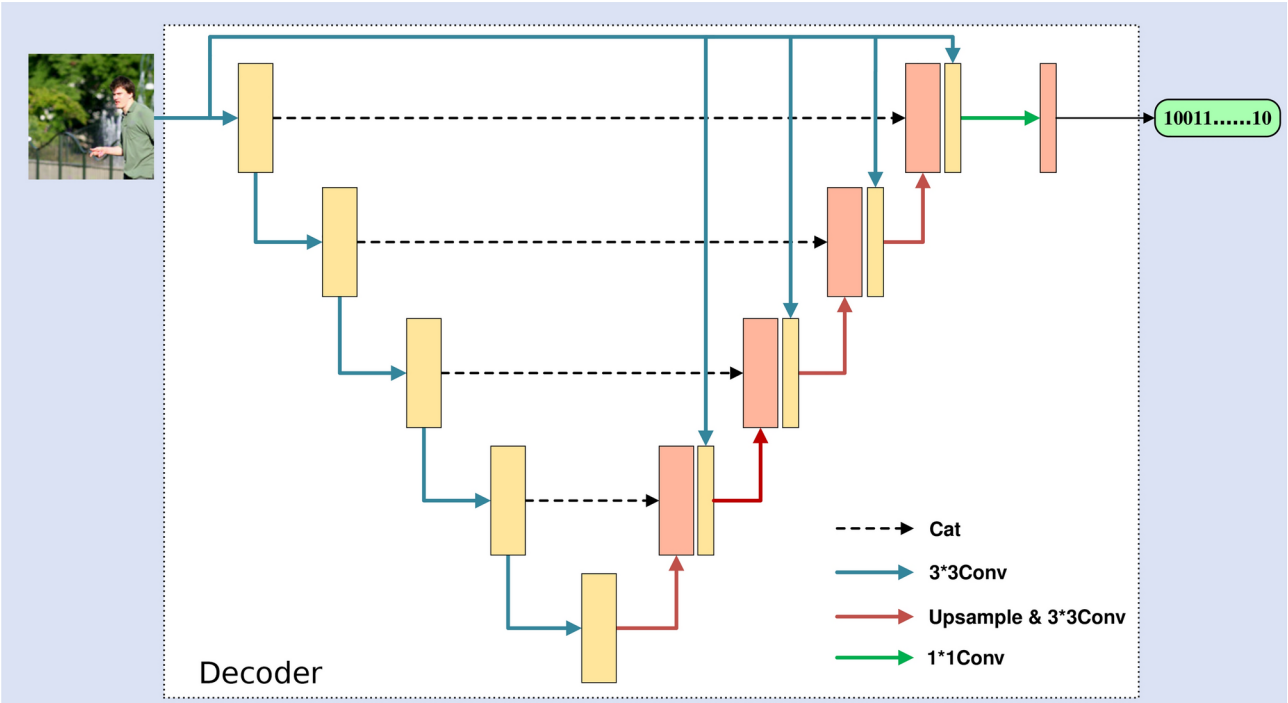

Fig. 4. Structure of decoder.

$$L_D = MSE(W_d, W) = MSE(D(\theta_D, I_e), W) \tag{5}$$

where W_d represents the watermark decoded by the decoder and W denotes the original watermark.

Discriminator

To achieve better visual quality in watermarked images, we incorporate a discriminator C into the model architecture for adversarial training with the encoder. The goal of C is to distinguish between the original cover image I_c and the watermarked image I_e. The encoder aims to generate watermarked images I_e that are visually indistinguishable from I_c while embedding the watermark, thereby "deceiving" the discriminator. Conversely, C is trained to detect I_e by identifying discrepancies between I_c and I_e. Through this adversarial process, the encoder produces watermarked images that retain high visual fidelity to the cover image. The discriminator C comprises five convolutional layers. The quality of I_e is improved by using the discriminator loss L_C to update the parameters θ_E:

$$L_C = \log(C(I_e)) = \log(C(E(\theta_E, I_c, W))) \tag{6}$$

Besides, C gives the correct classification of I_c and I_e by updating θ_C:

$$L_C = \log(1 - C(\theta_C, I_e)) = \log(1 - C(\theta_C, E(I_c, W))) \tag{7}$$

Loss function

The final loss of the entire network comprises the encoder, discriminator loss, and decoder loss mentioned earlier:

$$L = L_E + \lambda_4 L_D + \lambda_5 L_C = \lambda_1 L_{l2} + \lambda_1 L_{lpips} + \lambda_3 L_{ssim} + \lambda_4 L_D + \lambda_5 L_C \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ are set as 2.0, 1.5, 1, 1.5, 0.5 by default.

Experiments

This section first presents the details of the proposed implementation scheme and then compares the proposed scheme with the state-of-the-art watermarking schemes, including visual quality, resistance to simulation attacks, and resistance to screen-shooting attacks. Finally, ablation experiments were carried out to verify the effectiveness of the proposed scheme.

Implementation details

To train the proposed model, we randomly selected 10,000 images from the Mirflicker dataset[50], including 9000 images as training sets and 1000 images as validation sets. The split ratio between the training set and the validation set is 9 to 1. For easier comparison with other methods, all images were resampled to a resolution of 128×128 pixels, and the length of the watermark sequence was set to 64 bits. The framework was implemented using PyTorch[51], and the model was trained for 200 epochs on an NVIDIA Tesla P100 GPU with a batch size of 64. During the testing phase, 200 images were randomly chosen from the COCO dataset[52] to form the test set. The ratio of training set to test set is 45 to 1. For experiments requiring manual photography, 100 images were randomly selected from the test set and captured under various conditions. The screen used in the experiments was a "Redmi P24FBA-RA," and the capture device was a "realme X50 Pro Play." To ensure a fair comparison, we retrained each scheme on the same dataset.

Evaluation criteria

To ensure a fair evaluation of the performance of different schemes, the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) were utilized to assess the visual quality of the watermarked images. For evaluating the robustness of the watermark, the Bit Error Rate (BER) of the extracted watermark was employed as a metric. To maintain fairness, the BER values reported in the experiments for all schemes were those obtained without error correction coding.

Visual quality

In this section, we compare the visual quality of the proposed scheme with other methods, including StegaStamp[36], Steganogan[53], PIMOG[39], and CSRW[54]. We randomly selected 100 images from the test set and compared the average PSNR and SSIM values of the encoded images for different schemes; the results are presented in Fig. 5 and Table 2. As shown in Fig. 5, the images encoded by our proposed scheme are nearly indistinguishable from the original images. Furthermore, Table 2 reveals that the average PSNR value of our proposed scheme reaches 41.90 dB, and the average SSIM value reaches 0.99, indicating that our proposed scheme has high visual quality.

Simulation-based robustness test

We first evaluate the robustness of the proposed scheme under simulated attacks, where the watermarked image is directly attacked by some image manipulations. In addition to the distortions described in Section "Noise layer", salt and pepper noise, and JPEG compression are also included, as they typically occur during transmission in electronic channels. Specifically, the random disturbance range of the four corners of perspective transformation is set from -2 to 2. The intensity of point light source is set to 0.5, the a and b of the line light source are set to 0.2 and 0.5 respectively, the size of Gaussian blur kernel is set to 1, and the linear blur kernel is set to 3. The default value in[39] is adopted for the intensity of Moiré distortion. JPEG compression with quality factor QF = 50 and the

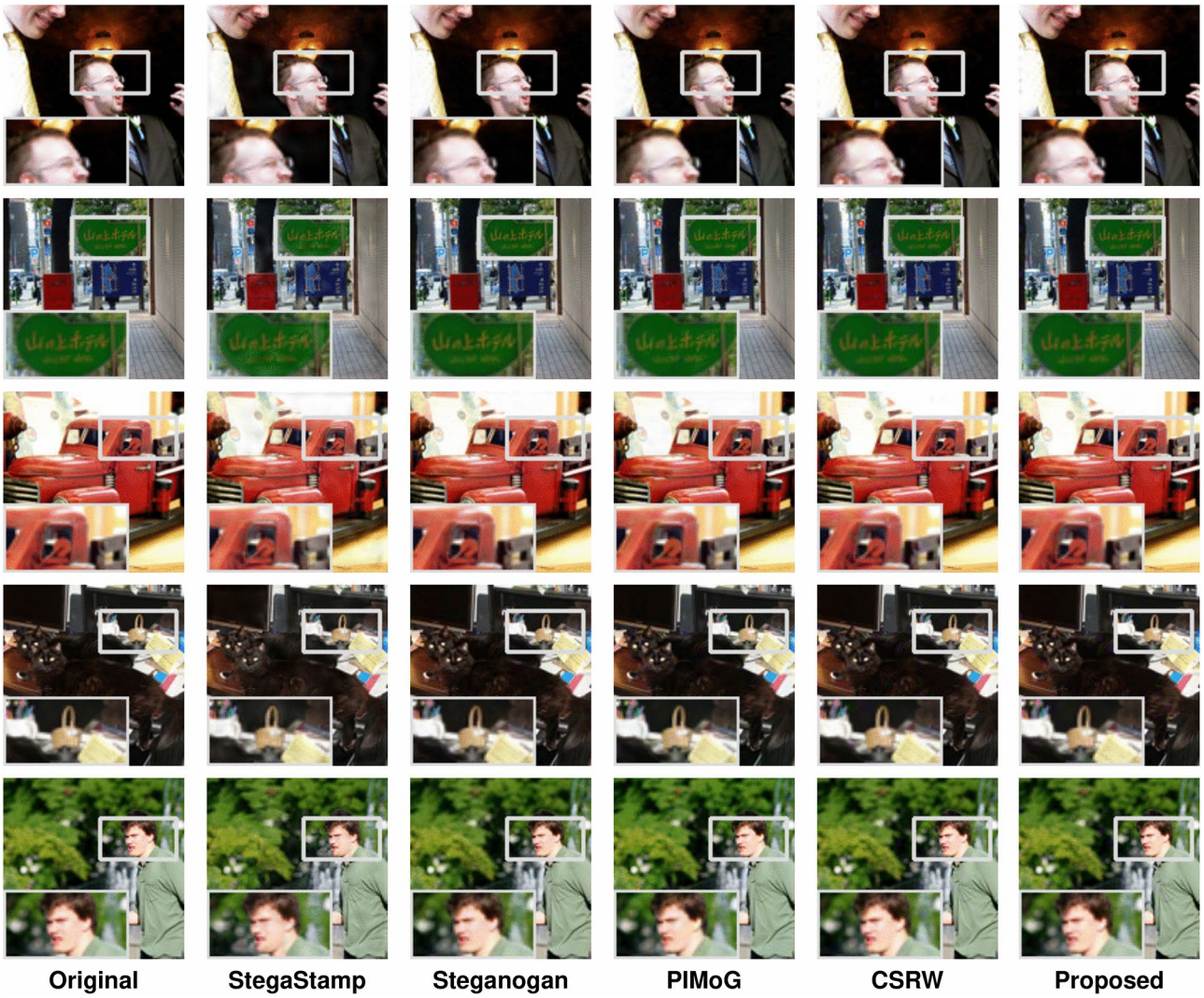

Fig. 5. The visual quality of watermarked images embedded with different methods.

Methods	StegaStamp[36]	Steganogan[53]	PIMoG[39]	CSRW[54]	Proposed
PSNR (dB)	29.86	37.24	37.49	35.47	41.90
SSIM	0.93	0.94	0.98	0.98	0.99

Table 2. The PSNR and SSIM values of different methods.

intensity of salt & pepper noise is set to 0.01. We compared the proposed scheme with other schemes, and the results are shown in Table 3. In Tables 3–8, [bold] values indicate the best-performing method under the same experimental conditions. Our proposed scheme achieves the best results in most of the attacks.

Screen-shooting robustness test

In this section, we evaluate the robustness of the proposed scheme against screen-shooting attacks. Specifically, we embedded a 64-bit watermark sequence into 100 randomly selected images from the test set and displayed these watermarked images on a screen, which were then captured under various conditions using a mobile phone. We retrained the other watermarking schemes on the same dataset to ensure a fair comparison. To standardize the preprocessing steps, the captured images were cropped and geometrically corrected using the same algorithm, as shown in Fig. 6. Subsequently, the watermark was extracted using a decoder without error correction mechanisms, and the BER of the extracted watermark was used as the evaluation metric.

Attacks	StegaStamp[23] (%)	PIMoG[26] (%)	CSRW[41] (%)	Proposed (%)
Perspective	9.81	0.62	0.15	**0.11**
Lighting	10.51	1.05	3.50	**0.52**
Blur	1.02	0.20	**0.10**	0.11
Moiré	11.25	0.45	3.42	**0.09**
salt&pepper	2.12	1.23	0.12	**0.05**
JPEG compression	2.10	1.20	0.09	**0.02**

Table 3. Comparison of BER under different image attacks.

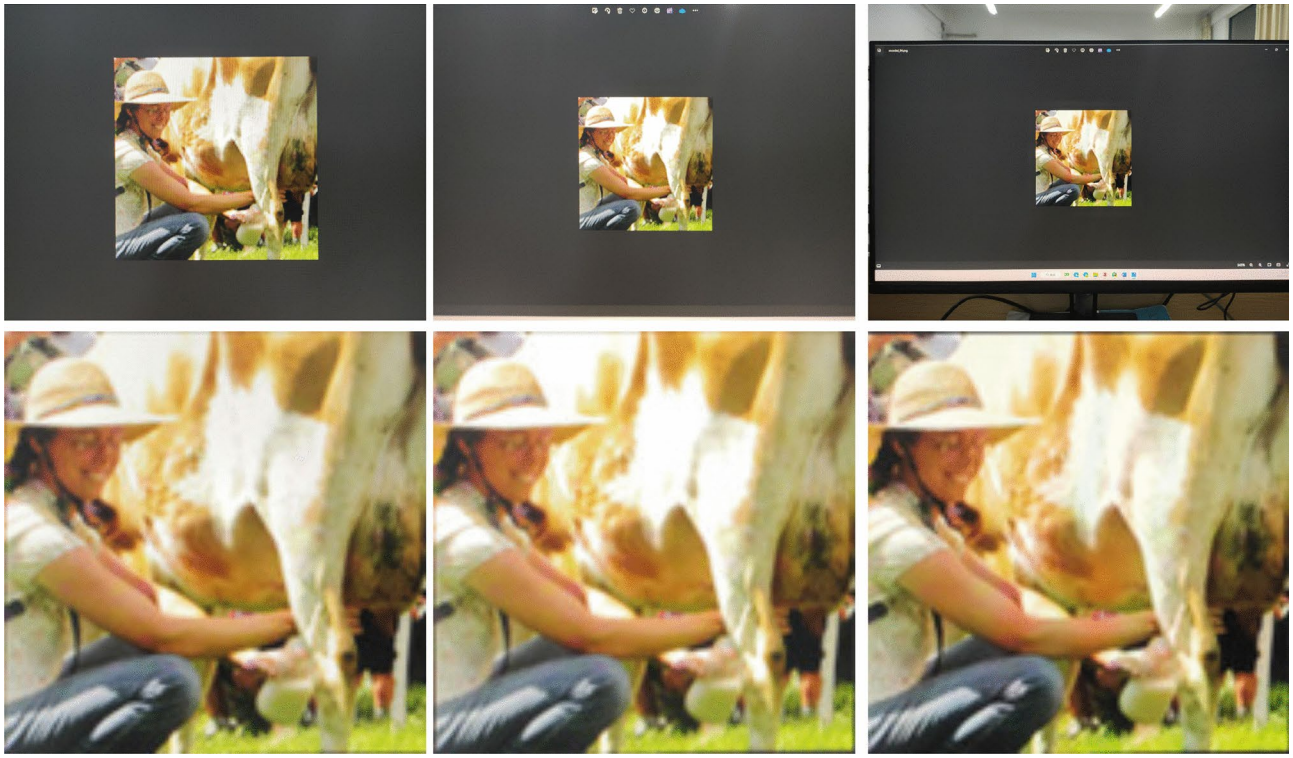

Fig. 6. The corrected screen-shooting image.

Distance (cm)	20	30	40	50
SSRW[23]	97.35%	95.88%	93.23%	87.81%
StegaStamp[36]	95.24%	92.50%	94.00%	95.25%
PIMoG[39]	96.50%	95.67%	94.67%	92.83%
CSRW[54]	98.00%	95.48%	96.75%	96.13%
WaveRecovery[55]	98.40%	97.17%	**97.10%**	96.83%
Proposed	**98.51%**	**97.55%**	95.58%	**97.27%**

Table 4. Extraction accuracy with different shooting distances.

Distance test

The distance from the camera to the screen can change in actual capture scenarios. To assess the robustness of the proposed scheme under such conditions, we conducted experiments at different capture distances. Watermarked images were exhibited on a screen and subsequently captured with a mobile phone, varying the capture distance between 20 and 50 cm. The experimental results are summarized in Table 4. As shown in Table 4, the proposed scheme maintains high accuracy across all tested distances, demonstrating its robustness to varying capture conditions.

Angles(°)	Left 15	Left 30	Left 45	Left 60	Left 70
SSRW[23]	90.87%	88.94%	64.21%	55.24%	58.67%
StegaStamp[36]	96.43%	92.48%	87.81%	71.25%	73.18%
PIMoG[39]	96.10%	81.42%	75.76%	65.49%	58.67%
CSRW[54]	98.45%	96.90%	95.48%	94.35%	95.70%
WaveRecovery[55]	97.19%	95.14%	96.63%	95.86%	95.66%
Proposed	**99.24%**	**97.97%**	**97.40%**	**97.82%**	**97.66%**

Table 5. Extraction accuracy with different horizontal shooting angles (Left).

Angles(°)	Right 15	Right 30	Right 45	Right 60	Right 70
SSRW[23]	88.48%	83.59%	72.07%	60.51%	59.57%
StegaStamp[36]	96.10%	94.20%	88.60%	72.41%	72.63%
PIMoG[39]	95.76%	82.08%	74.83%	64.58%	51.33%
CSRW[54]	99.15%	96.00%	95.95%	95.05%	94.50%
WaveRecovery[55]	97.30%	95.45%	95.09%	95.28%	95.19%
Proposed	**99.30%**	**96.10%**	**96.09%**	**97.97%**	**96.25%**

Table 6. Extraction accuracy with different horizontal shooting angles (Right).

Angles(°)	Up 15	Up 30	Up 45	Up 60	Up 70
SSRW[23]	87.69%	92.81%	72.07%	60.51%	59.57%
StegaStamp[36]	96.25%	93.25%	88.00%	76.30%	76.00%
PIMoG[39]	**98.96%**	**99.17%**	**98.75%**	60.67%	59.41%
CSRW[54]	98.36%	96.50%	95.58%	94.22%	95.75%
WaveRecovery[55]	97.20%	95.34%	96.46%	95.75%	95.12%
Proposed	98.70%	96.68%	98.10%	**97.92%**	**98.18%**

Table 7. Extraction accuracy with different vertical shooting angles (Up).

Angles(°)	Down 15	Down 30	Down 45	Down 60	Down 70
SSRW[23]	89.25%	85.84%	65.24%	56.21%	58.71%
StegaStamp[36]	96.00%	96.25%	86.75%	75.75%	75.75%
PIMoG[39]	99.16%	**98.96%**	**99.17%**	66.15%	54.61%
CSRW[54]	98.65%	96.40%	95.35%	94.08%	94.45%
WaveRecovery[55]	97.00%	95.55%	95.34%	95.20%	95.05%
Proposed	**99.22%**	96.72%	98.13%	**97.82%**	**97.55%**

Table 8. Extraction accuracy with different vertical shooting angles (Down).

Angle test

The capture angle significantly impacts the accuracy of watermark extraction. In real-world capture scenarios, the capturing device is often not perpendicular to the screen, leading to varying degrees of perspective distortion in the watermarked images. To evaluate the robustness of the proposed scheme under different capture angles, we conducted experiments with angles ranging from 15° to 70° in both horizontal and vertical directions. The results of these experiments are presented in Tables 5, 6, 7 and 8.

The experimental results show that the extraction accuracy of the proposed method is consistently higher than 95% in all tested conditions. In contrast, other methods exhibit varying degrees of performance degradation as the capture angle increases or the capture distance extends. For instance, PIMoG[39] experiences a sharp decline in accuracy when the capture angle exceeds 45°. Our proposed scheme, however, maintains higher robustness even at larger capture angles, demonstrating its superior resilience to perspective distortions.

Device test

The previously mentioned experiments were conducted using a "Redmi P24FBA-RA" monitor and a "realme X50 Pro Play" smartphone. Additional experiments were performed with diverse screens and mobile devices

Device	X50 pro play (%)	iPhone 15 Pro (%)	Xiaomi 13 (%)
Redmi P24FBA-RA	98.51	99.17	98.75
AOC Q24G2	99.15	99.54	99.16

Table 9. Extraction accuracy with different devices.

	PIMoG[39]	SSRIW[16]	Proposed
FLOPs ($\times 10^6$)	1.98	60.10	47.17
Params ($\times 10^3$)	3.57	68.81	50.55

Table 10. Comparisons of computational efficiency with other schemes.

to assess the proposed scheme's adaptability across hardware configurations. In these experiments, the capture distance was fixed at 20 cm, and both horizontal and vertical angles were set to 0°. For consistency, captured images were cropped and geometrically corrected using a standardized algorithm, without error correction coding. Table 9 summarizes the results. As shown in Table 9, the proposed method achieved high extraction accuracy across all device combinations, demonstrating strong adaptability.

Complexity analysis

The complexity of the proposed watermarking system can be analyzed from three main aspects: computational complexity, storage requirements, and time requirements.

Computational complexity

1. Encoder (Cross-Attention Mechanism): The encoder employs a multi-head cross-attention mechanism to embed the watermark into the cover image. The computational complexity of the cross-attention mechanism is primarily determined by the matrix multiplications between the query, key, and value matrices. Given that the image is divided into 256 patches (8×8 pixels each), and each patch has a dimension of 64, the computational complexity can be approximated as $O(n^{2d})$, where n is the number of patches (256) and d is the dimension of each patch (64). This results in a manageable computational load, as the operations are optimized for efficiency.
2. Decoder (Enhanced U-Net): The decoder is based on an enhanced U-Net architecture, which involves convolutional layers and skip connections. The computational complexity of the convolutional layers is $O(k^2 * C_{in} * C_{out} * H * W)$, where k is the kernel size, C_{in} and C_{out} are the input and output channels, and H and W are the height and width of the feature maps. The use of skip connections allows the decoder to efficiently reuse features from earlier layers, reducing redundant computations and maintaining a reasonable computational complexity.
3. Discriminator: The discriminator is a convolutional neural network with five layers, designed to distinguish between the original cover image and the watermarked image. Its computational complexity is relatively low compared to the encoder and decoder, as it processes images at a coarser level and focuses on high-level features.

Storage requirements

The proposed system has moderate storage requirements. The total number of parameters in the model, including the encoder, decoder, and discriminator, is approximately 50.55 thousand. This is relatively lightweight compared to other deep learning models, making it suitable for deployment in environments with limited storage resources.

Time requirements

1. Training Time: The model was trained for 200 epochs on an NVIDIA Tesla P100 GPU with a batch size of 64. The training process is feasible with modern GPU resources and can be completed within a reasonable timeframe.
2. Inference Time: The average encoding time of the proposed encoder is 0.0032 seconds, which is significantly faster than other methods. This makes the system suitable for real-time applications, where low latency is crucial.

To demonstrate that the proposed scheme is relatively lightweight, we compared it with the methods in[16,39]. Standard metrics for evaluating model complexity include FLOPs (Floating Point Operations) and Params (number of model parameters), where lower values indicate reduced memory usage and faster inference speed. These metrics are influenced by the input image size and watermark length. In this experiment, the input image size was set to 128×128, and the watermark length was 64 bits. We used the "thop" toolkit to calculate the FLOPs and Params, with results presented in Table 10. While the proposed model has higher computational complexity and more parameters than PIMoG[39] (which uses the CNNs), it is significantly more lightweight

	PIMoG[39]	SSRIW[16]	Proposed
Execution Time(s)	0.05	0.19	0.0032

Table 11. Comparisons of average execution time with other schemes.

	PSNR (dB)	SSIM
w/ cross attention	41.90	0.99
w/o cross attention	39.09	0.99

Table 12. The PSNR and SSIM values of different models.

than the architecture in[16]. Additionally, we measured the average encoding time of different encoders. As shown in Table 11, despite the lower computational complexity of[39], our proposed scheme still offers a notable speed advantage.

In summary, the proposed watermarking system has a manageable computational complexity, moderate storage requirements, and efficient time requirements. The use of a multi-head cross-attention mechanism and an enhanced U-Net architecture ensures that the system can effectively embed and extract watermarks while maintaining high visual quality and robustness. The system is well-suited for practical applications, particularly those requiring real-time processing and deployment in resource-constrained environments.

Ablation study

Cross attention

The proposed scheme replaces the CNNs with a multi-head cross-attention mechanism for the purpose of watermark embedding. To evaluate the performance of the multi-head cross-attention encoder as compared to the CNNs encoder, we substituted our model's encoder with the StegaStamp model's encoder and retrained it on the same dataset. The comparison results are presented in Table 12. Since the SSIM value was used as one of the loss functions to constrain the encoder, both schemes achieved an SSIM value of 0.99. However, the cross-attention scheme outperformed the CNN-based scheme in terms of PSNR. This is because the cross-attention mechanism allows the model to effectively extract global features of both the cover image and the watermark, thereby identifying visually imperceptible regions for robust watermark embedding.

SSIM LOSS

To achieve better-encoded image quality, we incorporated the SSIM as one component of the hybrid loss function for constraining the encoder. To validate this design choice, we trained two models: one with the SSIM loss and another without it. The results are shown in Fig. 7. As illustrated, the model without the SSIM loss produced images with noticeable visual distortions, achieving an average PSNR of only 31.45 dB on the test set.

Conclusion

In this paper, we present a novel end-to-end screen-shooting resistant watermarking method. Drawing on the ViTs approach, the scheme segments both the cover and watermark images and applies a multi-head cross-attention mechanism for embedding. This enables the model to analyze the entire image and strategically select embedding positions. Additionally, we enhanced the U-Net architecture and utilized it as the decoder. By integrating skip connections from various scales of the initial input image into the expansive path of the network, we mitigate information loss caused by repeated downsampling, thus improving watermark extraction accuracy. Through extensive experiments, the proposed scheme has proven effective across a range of conditions and demonstrates superior robustness compared to existing methods. A limitation of our current scheme is the concentration of watermark embedding modifications in the image edges. This means that incomplete capture of the watermarked image significantly degrades extraction accuracy. To address this, future work will refine the embedding method. One potential solution involves dividing the watermark into multiple segments and embedding them repeatedly across different image regions. While this may reduce watermark capacity, it could resolve issues arising from partial image captures.

Fig. 7. Encoded images with and without ssim loss.

Data availability

The images used in the study were acquired from public databases COCO[52] and Mirflicker[50]. Data publicly available in a repository: https://cocodataset.org/#download and https://press.liacs.nl/mirflickr/.

Received: 11 December 2024; Accepted: 2 May 2025
Published online: 16 May 2025

References

1. Hameed, M. A., Abdel-Aleem, O. A. & Hassaballah, M. A secure data hiding approach based on least-significant-bit and nature-inspired optimization techniques. *J. Ambient Intell. Hum. Comput.* **14**(5), 4639–4657. https://doi.org/10.1007/s12652-022-04366-y (2023).
2. Hassaballah, M., Hameed, M.A., & Alkinani, M.H. Introduction to digital image steganography. In *Digital Media Steganography*, 1–15. (Elsevier, 2020). https://doi.org/10.1016/B978-0-12-819438-6.00009-8
3. Hassaballah, M., Hameed, M.A., Aly, S., & AbdelRady, A. A color image steganography method based on ADPVD and HOG techniques. In *Digital Media Steganography*, 17–40. (Elsevier, 2020). https://doi.org/10.1016/B978-0-12-819438-6.00010-4
4. Hassaballah, M., Hameed, M. A., Awad, A. I. & Muhammad, K. A novel image steganography method for industrial internet of things security. *IEEE Trans. Ind. Inform.* **17**(11), 7743–7751. https://doi.org/10.1109/TII.2021.3053595 (2021).
5. Hameed, M. A., Hassaballah, M., Aly, S. & Awad, A. I. An adaptive image steganography method based on histogram of oriented gradient and PVD-LSB techniques. *IEEE Access* **7**, 185189–185204. https://doi.org/10.1109/ACCESS.2019.2960254 (2019).
6. Hameed, M. A., Hassaballah, M., Abdelazim, R. & Sahu, A. K. A novel medical steganography technique based on adversarial neural cryptography and digital signature using least significant bit replacement. *Int. J. Cognit. Comput. Eng.* **5**, 379–397. https://doi.org/10.1016/j.ijcce.2024.08.002 (2024).
7. Solak, S., Abdirashid, A. M., Adjevi, A. & Sahu, A. K. Robust data hiding method based on frequency coefficient variance in repetitive compression. *Eng. Sci. Technol. Int. J.* **56**, 101756. https://doi.org/10.1016/j.jestch.2024.101756 (2024).
8. Sahu, A. K. & Swain, G. A novel multi stego-image based data hiding method for gray scale image. *Pertanika J. Sci. Technol.* **27**(2), 753–768 (2019).
9. Mali, S. D. & Agilandeeswari, L. Deepsecure watermarking: Hybrid attention on attention net and deep belief net based robust video authentication using quaternion curvelet transform domain. *Egypt. Inform. J.* **27**, 100514. https://doi.org/10.1016/j.eij.2024.100514 (2024).
10. Mali, S. D. & Agilandeeswari, L. Non-redundant shift-invariant complex wavelet transform and fractional gorilla troops optimization-based deep convolutional neural network for video watermarking. *J. King Saud Univ. Comput. Inf. Sci.* **35**(8), 101688. https://doi.org/10.1016/j.jksuci.2023.101688 (2023).
11. Loganathan, A. & Kaliyaperumal, G. An adaptive HVS based video watermarking scheme for multiple watermarks using bam neural networks and fuzzy inference system. *Expert Syst. Appl.* **63**, 412–434. https://doi.org/10.1016/j.eswa.2016.05.019 (2016).

12. Li, Y., Liao, X. & Wu, X. Screen-shooting resistant watermarking with grayscale deviation simulation. *IEEE Trans. Multimedia* https://doi.org/10.1109/TMM.2024.3415415 (2024).
13. Cao, F. et al. Universal screen-shooting robust image watermarking with channel-attention in DCT domain. *Expert Syst. Appl.* **238**, 122062. https://doi.org/10.1016/j.eswa.2023.122062 (2024).
14. Luo, T. et al. Wformer: A transformer-based soft fusion model for robust image watermarking. *IEEE Trans. Emerg. Top. Comput. Intell.* https://doi.org/10.1109/TETCI.2024.3386916 (2024).
15. Dosovitskiy, A. An image is worth 16×16 words: Transformers for image recognition at scale (2020). https://doi.org/10.48550/arXiv.2010.11929. arXiv:2010.11929
16. Dasgupta, A., & Zhong, X. Robust image watermarking based on cross-attention and invariant domain learning. In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1125–1132 (2023). https://doi.org/10.1109/CSCI62032.2023.00185
17. Van Schyndel, R.G., Tirkel, A.Z., & Osborne, C.F. A digital watermark. In *Proceedings of 1st International Conference on Image Processing. IEEE*, vol. 2, 86–90 (1994). https://doi.org/10.1109/ICIP.1994.413536
18. Bender, W., Gruhl, D., Morimoto, N. & Lu, A. Techniques for data hiding. *IBM Syst. J.* **35**(3.4), 313–336. https://doi.org/10.1147/sj.353.0313 (1996).
19. Solachidis, V. & Pitas, L. Circularly symmetric watermark embedding in 2-d DFT domain. *IEEE Trans. Image Process.* **10**(11), 1741–1753. https://doi.org/10.1109/ICASSP.1999.757589 (2001).
20. Guan, H., Zeng, Z., Liu, J., & Zhang, S. A novel robust digital image watermarking algorithm based on two-level DCT. In *2014 International Conference on Information Science, Electronics and Electrical Engineering. IEEE*, vol. 3, 1804–1809 (2014). https://doi.org/10.1109/InfoSEEE.2014.6946233
21. Dugad, R., Ratakonda, K., & Ahuja, N. A new wavelet-based scheme for watermarking images. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269). IEEE*, vol. 2, 419–423 (1998). https://doi.org/10.1109/ICIP.1998.723406
22. Liu, R. & Tan, T. An SVD-based watermarking scheme for protecting rightful ownership. *IEEE Trans. Multimedia* **4**(1), 121–128. https://doi.org/10.1109/6046.985560 (2002).
23. Fang, H., Zhang, W., Zhou, H., Cui, H. & Yu, N. Screen-shooting resilient watermarking. *IEEE Trans. Inf. Forensics Secur.* **14**(6), 1403–1418. https://doi.org/10.1109/TIFS.2018.2878541 (2018).
24. Chen, W. et al. Screen-cam robust image watermarking with feature-based synchronization. *Appl. Sci.* **10**(21), 7494. https://doi.org/10.3390/app10217494 (2020).
25. Amiri, S. H. & Jamzad, M. Robust watermarking against print and scan attack through efficient modeling algorithm. *Signal Process. Image Commun.* **29**(10), 1181–1196. https://doi.org/10.1016/j.image.2014.07.004 (2014).
26. Huang, Y., Niu, B., Guan, H. & Zhang, S. Enhancing image watermarking with adaptive embedding parameter and PSNR guarantee. *IEEE Trans. Multimedia* **21**(10), 2447–2460. https://doi.org/10.1109/TMM.2019.2907475 (2019).
27. Zhu, J.: Hidden: Hiding data with deep networks. arXiv:1807.09937 (2018).
28. Chen, B., Wu, Y., Coatrieux, G., Chen, X. & Zheng, Y. JSNet: A simulation network of JPEG lossy compression and restoration for robust image watermarking against JPEG attack. *Comput. Vis. Image Understand.* **197**, 103015. https://doi.org/10.1109/CVPR.2018.00657 (2020).
29. Ahmadi, M., Norouzi, A., Karimi, N., Samavi, S. & Emami, A. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Syst. Appl.* **146**, 113157. https://doi.org/10.1016/j.eswa.2019.113157 (2020).
30. Liu, Y., Guo, M., Zhang, J., Zhu, Y., & Xie, X. A novel two-stage separable deep learning framework for practical blind watermarking. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1509–1517 (2019). https://doi.org/10.1145/3343031.3351025
31. Kandi, H., Mishra, D. & Gorthi, S. R. S. Exploring the learning capabilities of convolutional neural networks for robust image watermarking. *Comput. Secur.* **65**, 247–268. https://doi.org/10.1016/j.cose.2016.11.016 (2017).
32. Fang, H. et al. Deep template-based watermarking. *IEEE Trans. Circuits Syst. Video Technol.* **31**(4), 1436–1451. https://doi.org/10.1109/TCSVT.2020.3009349 (2020).
33. Palani, A. & Loganathan, A. Semi-blind watermarking using convolutional attention-based turtle shell matrix for tamper detection and recovery of medical images. *Expert Syst. Appl.* **238**, 121903. https://doi.org/10.1016/j.eswa.2023.121903 (2024).
34. Fang, H. et al. De-end: Decoder-driven watermarking network. *IEEE Trans. Multimedia* **25**, 7571–7581. https://doi.org/10.1109/TMM.2022.3223559 (2022).
35. Wengrowski, E., & Dana, K. Light field messaging with deep photographic steganography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1515–1524 (2019). https://doi.org/10.1109/CVPR.2019.00161
36. Tancik, M., Mildenhall, B., & Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2117–2126 (2020). https://doi.org/10.1145/3319819.3326867
37. Jia, J., Gao, Z., Zhu, D., Min, X., Zhai, G., & Yang, X. Learning invisible markers for hidden codes in offline-to-online photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2273–2282 (2022). https://doi.org/10.1109/CVPR52688.2022.00231
38. Lu, J., Ni, J., Su, W., & Xie, H. Wavelet-based CNN for robust and high-capacity image watermarking. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6 (2022). IEEE. https://doi.org/10.1109/ICME52920.2022.9859725
39. Fang, H., Jia, Z., Ma, Z., Chang, E.-C., & Zhang, W. PIMoG: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2267–2275 (2022). https://doi.org/10.1145/3503161.3548049
40. Qin, C. et al. Print-camera resistant image watermarking with deep noise simulation and constrained learning. *IEEE Trans. Multimedia* https://doi.org/10.1109/TMM.2023.3293272 (2023).
41. Vaswani, A. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017). https://doi.org/10.48550/arXiv.1706.03762
42. Li, J., Zhang, H., Wang, J., Xiao, Y. & Wan, W. Orientation-aware saliency guided jnd model for robust image watermarking. *IEEE Access* **7**, 41261–41272. https://doi.org/10.1109/ACCESS.2019.2904272 (2019).
43. Eerapu, K. K., Lal, S. & Narasimhadhan, A. O-segnet: Robust encoder and decoder architecture for objects segmentation from aerial imagery data. *IEEE Trans. Emerg. Top. Comput. Intell.* **6**(3), 556–567. https://doi.org/10.1109/TETCI.2020.3045485 (2021).
44. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., & Torr, P.H. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6881–6890 (2021). https://doi.org/10.1109/CVPR46437.2021.00681
45. Lee, S.-G., Kim, E., Bae, J. S., Kim, J. H. & Yoon, S. Robust end-to-end focal liver lesion detection using unregistered multiphase computed tomography images. *IEEE Trans. Emerg. Top. Comput. Intell.* **7**(2), 319–329. https://doi.org/10.1109/TETCI.2021.3132382 (2021).
46. Barkur, R., Suresh, D., Lal, S., Reddy, C. S. & Diwakar, P. Rscdnet: A robust deep learning architecture for change detection from bi-temporal high resolution remote sensing images. *IEEE Trans. Emerg. Top. Comput. Intell.* **7**(2), 537–551. https://doi.org/10.1109/TETCI.2022.3230941 (2022).
47. Aberna, P. & Agilandeeswari, L. Optimal semi-fragile watermarking based on maximum entropy random walk and swin transformer for tamper localization. *IEEE Access* https://doi.org/10.1109/ACCESS.2024.3370411 (2024).
48. Palani, A. & Loganathan, A. Multi-image feature map-based watermarking techniques using transformer. *Int. J. Electr. Electron. Res.* **11**, 339–344. https://doi.org/10.37391/ijeer.110214 (2023).

49. Ronneberger, O., Fischer, P., & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. (Springer, 2015). https://doi.org/10.1007/978-3-319-24574-4_28
50. Huiskes, M.J., & Lew, M.S. The MIR flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, 39–43 (2008). https://doi.org/10.1145/1460096.1460104
51. Collobert, R., Kavukcuoglu, K., & Farabet, C. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop* (2011). https://doi.org/10.1145/2906944.2906957
52. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. (Springer, 2014). https://doi.org/10.1109/ECCV.2014.87
53. Zhang, K.A., Cuesta-Infante, A., Xu, L., & Veeramachaneni, K. Steganogan: High capacity image steganography with GANs. arXiv:1901.03892. arXiv:1908.01073 (2019).
54. He, M., Feng, B., Guo, Y., Weng, J. & Lu, W. Camera-shooting resilient watermarking on image instance level. *IEEE Trans. Circuits Syst. Video Technol.* https://doi.org/10.1109/TCSVT.2024.3411816 (2024).
55. Fu, L., Liao, X., Guo, J., Dong, L. & Qin, Z. Waverecovery: Screen-shooting watermarking based on wavelet and recovery. *IEEE Trans. Circuits Syst. Video Technol.* https://doi.org/10.1109/TCSVT.2024.3510355 (2024).

Acknowledgements

This study was funded by the Shandong Province Natural Science Foundation (No. ZR2022MF277), Shandong Province Key R&D Plan (Soft Science) Project (No.2023RKL01003), and the joint Fund of Natural Science Foundation of Shandong province (ZR202209070011).

Author contributions

Lianshan Liu: Conceptualization, Writing-review and editing, Formal analysis, Methodology, Investigation, Funding acquisition. Peng Xu: Conceptualization, Writing-original draft, Writing-review and editing, Formal analysis, Methodology, Investigation, Software. Qianwen Xue: Investigation, Validation, Resources, Supervision, Project administration, Funding acquisition.

Declarations

Competing interests

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Additional information

Correspondence and requests for materials should be addressed to L.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.